Nonlinear Asymptotic Integration Algorithms for One-Dimensional Autonomous Dissipative First-Order ODEs

Henry W. Haslach, Jr.
Department of Mechanical Engineering
University of Maryland
College Park, Maryland 20742

Alan D. Freed
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Kevin P. Walker
Engineering Science Software, Inc.
Smithfield, Rhode Island 02917

March 28, 1994

Abstract

Nonlinear asymptotic integrators are applied to one-dimensional, nonlinear, autonomous, dissipative, ordinary differential equations. These integrators, including a one-step explicit, a one-step implicit, and a one- and a two-step midpoint algorithm, are designed to follow the asymptotic behavior of a system approaching a steady state. The methods require that the differential equation be written in an particular asymptotic form. This is always possible for a one-dimensional equation with a globally asymptotic steady state. In this case, conditions are obtained to guarantee that the implicit algorithms are well defined. Further conditions are determined for the implicit methods to be contractive. These methods are all first order accurate, while under certain conditions the midpoint algorithms may also become second order accurate. The stability of each method is investigated and an estimate of the local error is provided.

1 Introduction

Modeling non-equilibrium material behavior, such as viscoplasticity or chemical reactions, involves dissipation according to the second law of thermodynamics. The processes of primary interest are modeled by dissipative, ordinary differential equations with at least one asymptotically stable steady state. Most classical algorithms for the numerical solution of systems of autonomous, ordinary differential equations are inefficient when used to predict the response of a material body as it approaches a steady state. The small step sizes that are usually required drastically increase the computational expense and time required to produce accurate solutions.

Nonlinear asymptotic algorithms, which are intended to be more efficient in tracking asymptotic behavior, can be defined by several types of asymptotic approximation. These algorithms were originally proposed for viscoplastic evolution problems by WALKER and FREED (1991); such evolution is driven by thermodynamic dissipation. The primary goal of this paper is to investigate the consistency and stability of these algorithms and to determine when those that are implicitly defined have a unique solution.

Dissipative evolution equations define a non-Hamiltonian dynamical system. In a Hamiltonian system, the volume in state (phase) space is conserved over time. A dissipative system is one in which the volume in state space, as opposed to the material, contracts. The instantaneous rate of change of the local volume is the divergence of the velocity field. This could vary from point to point and may even be positive. LICHTENBERG and LIEBERMAN (1992) call a system dissipative if the local volume averaged over an orbit in state space contracts. In continuum mechanics, the condition that the trace of the Jacobian matrix of $\dot{\mathbf{X}} = F(\mathbf{X})$ is negative implies that the density of the continua is increasing and, thus, that the volume is decreasing over time. Therefore solutions must behave asymptotically.

A solution algorithm is most effective when it reflects the behavior of the original differential equation. The asymptotic algorithms are particularly appropriate for dissipative systems. A one-dimensional system, $\dot{x} = f(x)$, is dissipative if $df/dx < 0$ for all x. If such a system has a unique steady state, it is globally asymptotically stable; the exact solutions to the differential equation are asymptotic to the steady state. The asymptotic algorithms of WALKER and FREED (1991) require that the differential equation be written in a special form. This is always possible for dissipative systems with a globally asymptotic steady state. Most of the results presented in this paper are for such equations.

Two major issues are considered for the asymptotic algorithms applied to this family. First, the method must be well defined. In other words, there must exist a unique solution to the difference equation. If the algorithm is implicit, there are two numerical methods in common usage to solve the nonlinear difference equation for the value at the next time step, either the Newton-Raphson method or the method of successive approximations. The Newton-Raphson method, when applied to solve these equations, must approach the desired root of the difference equation. In general, restrictions on the step size are required to guarantee such convergence, see for example the Kantorovich theorem (RALL 1974). Furthermore, if the equation $f(x) = 0$ has several roots, it is difficult to guarantee that the Newton-Raphson method will select the correct root. For example, for the equation $x^3 - x = 0$, perturbations of a starting value near the inflection point can cause the Newton-Raphson method to converge to any of the three roots. One solution is to apply the algorithms only in situations which guarantee that the implicit difference equation has a unique root.

Second, any useful algorithm must converge to the exact solution for the given nonlinear differential equation. Convergence to a limit is controlled by the stability of the algorithm. The limit is the exact solution if the algorithm is first-order accurate. These two criteria are verified under some limits on the step size for the asymptotic algorithms applied to the family of dissipative differential equations. The generalized midpoint asymptotic algorithms discussed are shown to be second order accurate under certain conditions.

The classic ideas of stability, A-stability, L-stability, B-stability, and G-stability, all were defined in terms of systems of linear equations. Stability is largely determined by the location of the eigenvalues of the Jacobian of the linearized system in the complex plane. There is still no general agreement on a proper definition of stability for algorithms nonlinear in the step size when applied to nonlinear systems of ordinary differential equations. Some possible definitions are considered here.

The asymptotic algorithms are defined in Section 2. Dissipative, one-dimensional, ordinary differential equations are defined in Section 3, along with the special formulation required to write the difference equation for the asymptotic integrators. The conditions that these algorithms should satisfy are summarized in Section 4. The final four sections develop the properties of the asymptotic explicit, implicit, two-step midpoint, and the one-step midpoint algorithms.

2 The Asymptotic Algorithms

An asymptotic algorithm is intended to accurately model the behavior of a solution as it approaches a steady state. In one dimension, then, the algorithms can be used to integrate autonomous differential equations $\dot{x} = f(x)$ such that the equation, $f(x) = 0$, has at least one root that is an asymptotically stable steady-state for the evolving system.

The asymptotic integrators described in WALKER and FREED (1991) require that the one-dimensional, ordinary differential equation be written in the form

$$\dot{x} = c(x)[a(x) - x]. \tag{1}$$

WALKER and FREED derived a variety of asymptotic approximations by first applying an integrating factor to Eqn. (1).

$$\begin{aligned} x(t+h) &= \exp\left[-\int_t^{t+h} c(x(\eta))\, d\eta\right] x(t) \\ &+ \int_t^{t+h} \exp\left[-\int_\eta^{t+h} c(x(\xi))\, d\xi\right] c(x(\eta))a(x(\eta))\, d\eta. \end{aligned} \tag{2}$$

The first-order asymptotic integrators are obtained by approximating the integrals in Eqn. (2). To shorten the notation, put $c_n \equiv c(x_n)$, etc., and similarly for $a(x)$, where x_n is the approximation to the solution at the n^{th} time step, $x(t_n)$. In particular, one obtains the *first-order, asymptotic, forward integrator,* by approximating $c(x)$ by the constant c_n and $a(x)$ by a_n.

$$x_{n+1} = x_n + \left(1 - e^{-c_n h}\right)\left(a_n - x_n\right) , \tag{3}$$

and the *first-order, asymptotic, backward integrator,* by approximating $c(x)$ by the constant c_{n+1} and $a(x)$ by a_{n+1}.

$$x_{n+1} = x_n + \left(1 - e^{-c_{n+1} h}\right)\left(a_{n+1} - x_n\right) . \tag{4}$$

A θ-midpoint integrator—first presented in FREED *et al.* (1992)—is equivalent to approximating $c(x)$ by the constant $c_{n+\theta}$ and $a(x)$ by $a_{n+\theta}$, resulting in the *first-order, two-step, asymptotic, θ-midpoint integrator,*

$$x_{n+1} = x_n + \left(1 - e^{-c_{n+\theta} h}\right)\left(a_{n+\theta} - x_n\right) , \tag{5}$$

where

$$x_{n+\theta} = x_n + [1 - \exp(-\theta c_{n+\theta} h)](a_{n+\theta} - x_n),$$

with $c_{n+\theta} \equiv c(x_{n+\theta})$ and $a_{n+\theta} \equiv a(x_{n+\theta})$.

A one-step θ-midpoint integrator is developed below. While the two-step θ-midpoint integrator can be viewed as first applying the implicit and then the explicit asymptotic algorithm, the one-step method is obtained by first applying the explicit and then the implicit asymptotic algorithm. It is the *first-order, one-step, asymptotic, θ-midpoint integrator,*

$$x_{n+1} = x_n + \left(1 - e^{-[(1-\theta)c_n + \theta c_{n+1}]h}\right)(a_n - x_n) + \left(1 - e^{-\theta c_{n+1} h}\right)(a_{n+1} - a_n). \tag{6}$$

Whenever $\theta = 0$, these reduce to the forward integrator, Eqn. (3); likewise, whenever $\theta = 1$, they reduce to the backward integrator, Eqn. (4).

These asymptotic algorithms each produce the exact solution for a one-dimensional, linear, ordinary differential equation, $\dot{x} = c(a - x)$, where a and c are constants. The exact solution is

$$x(t) = [x(0) - a]e^{-ct} + a,$$

while the algorithms yield

$$x_{n+1} = [x_n - a]e^{-ch} + a.$$

On the other hand, the classical one-step forward and backward Euler methods do not yield the exact solution in this simplest of all situations. Therefore, the asymptotic algorithms can be expected to be an improvement over linear algorithms.

3 Formulation of the System of Equations

For one-dimensional equations, the definition of the algorithms requires that the equation be written in terms of an asymptote, $a(x)$, and a coefficient, $c(x)$. Conditions are required for the autonomous differential equation to be rewritten in the form of Eqn (1), for those cases when it is not already presented in this form. Irrespective of the form of a one-dimensional, ordinary differential equation, it can only have monotonic solutions.

Lemma 3.1 *All solutions to an autonomous, single-dimensional, differential equation, $dx/dt = f(x)$ are monotonic.*

Proof: If any non-constant solution $x(t)$ is not monotonic, then there is a t^* such that $x(t^*) = x^*$ for some x^* such that $\dot{x} = f(x^*) = 0$. A root, x^*, of $f(x) = 0$ produces the constant solution $x(t) = x^*$ when the initial condition is $x(0) = x^*$. These two solutions passing through (t^*, x^*) violate the existence and uniqueness theorem for differential equations.

This reflects the fact that a one-dimensional system can only have stable or unstable, isolated, steady states. Periodic orbits are possible in two or higher dimensions, and chaotic behavior is possible in three or higher dimensions.

A solution, $x(t)$, approaches steady state if $\lim_{t\to\infty} dx/dt = 0$. All such solutions to $dx/dt = f(x)$ are asymptotic to a constant solution $x(t) = x^*$, where $f(x^*) = 0$. The sign of df/dx determines which constant solutions are asymptotes for other solutions, i. e. those which are locally asymptotically stable. Only if $df/dx < 0$ at x^* satisfying $f(x^*) = 0$ is the root x^* a locally asymptotic steady state. In the formulation, $dx/dt = f(x) = c(x)[a(x) - x]$, for $c(x) > 0$, the term $a(x)$, the asymptote function, determines how many and which steady states exist, i. e. where $a(x) - x = 0$. For example, if $a(x) = 0$, then there is a unique steady state, and it is globally asymptotically stable.

LAMBERT (1991) makes the following definitions.

Definition 3.2 *The system $dx/dt = f(x)$ is dissipative in a region $R \times [a, b]$ if the norm*

$$\langle f(x) - f(y), x - y \rangle \leq 0$$

for all $x, y \in R$ and all $t \in [a, b]$.

Definition 3.3 *The system $dx/dt = f(x)$ is contractive in the interval $[a, b]$ if two solutions $x(t)$ and $y(t)$ with different initial conditions satisfy $|x(t_2) - y(t_2)| \leq |x(t_1) - y(t_1)|$ for all t_1, t_2 satisfying $a \leq t_1 \leq t_2 \leq b$.*

Any dissipative system is contractive because the difference $|x(t) - y(t)|$ is a non-increasing function of time when $\langle f(x) - f(y), x - y \rangle \leq 0$. Consequently, any one-dimensional system with a globally asymptotic steady-state is contractive.

Lemma 3.4 *If the one-dimensional system $dx/dt = f(x)$ has a globally asymptotic steady state, it is contractive.*

Proof: $dx/dt = f(x)$ has a globally asymptotic steady state if $df/dx < 0$ for all x. Therefore $x > y$ implies $f(x) < f(y)$. In the one-dimensional case, the norm is

$$[f(x) - f(y)](x - y) < 0.$$

Therefore the system is dissipative and also contractive.

This result suggests that to reflect the behavior of the differential equation, the integrator should also be contractive. This idea will be discussed in general in the next section, and for the specific algorithms in later sections.

The expression of a given system of ordinary differential equations in the form of Eqn. (1) is not likely to be unique. The two extremes of the range of equivalent expressions are obtained from putting all the nonlinearity in the coefficient $c(x)$ by making the asymptote zero, or from putting all the nonlinearity in the asymptote by making the coefficient $c(x)$ a constant. Numerical experimentation suggests that the more efficient technique is to put all the nonlinearity in the coefficient $c(x)$, whenever possible.

In a one-dimensional system, the existence of a globally asymptotic steady state is equivalent to being able to put all the nonlinearity in the coefficient, $c(x)$. In higher dimensional systems, being able to put all the nonlinearity in the coefficients, $c(x)$, implies that the system has a unique globally asymptotic steady state, but the converse is not true. If necessary, a coordinate change guarantees that $x = 0$ is the stable state.

Lemma 3.5 *Let $dx/dt = f(x)$ be such that $x = 0$ is the unique root of $f(x)$ and $df/dx < 0$ for all x, so that $x = 0$ is a globally asymptotic steady-state. Then the differential equation can be written in the form $dx/dt = c(x)(0 - x)$, where $a(x) = 0$. Conversely, if $dx/dt = -c(x)x$, for $c(x) > 0$, then $x = 0$ is a globally asymptotically stable steady-state.*

Proof: Define the coefficient, $c(x)$, by

$$c(x) = \begin{cases} -f(x)/x & \text{if} \quad x \neq 0 \\ -\frac{df}{dx}\Big|_{x=0} & \text{if} \quad x = 0 \end{cases}$$

Then $c(x)$ is continuous by L'Hospital's rule. It is positive for all x since the fact that $x = 0$ is globally asymptotic implies that $f(x) > 0$ for $x < 0$ and $f(x) < 0$ for $x > 0$.

Conversely, the Lyapunov function $V(x) = x^2$ shows that $x = 0$ is a globally asymptotic steady-state because $dx/dt = 0$ at $x = 0$ and $dV/dt = 2x(dx/dt) = -2c(x)x^2 < 0$ when $c(x) > 0$.

Example 3.6 *That the formulation is not unique can be shown by the differential equation, $dx/dt = -x^3 - x$, which has a globally asymptotically stable equilibrium at $x = 0$. It is in the form of (1) with $c(x) = 1$ and $a(x) = -x^3$. It can also can be written as $dx/dt = -(x^2+1)x$ so that $c(x) = x^2 + 1 > 0$, and $a(x) = 0$. Notice that the unique minimum of $c(x)$ occurs at the steady state in this example.*

Example 3.7 *The equation $\dot{x} = 1 - x^3$ has a globally asymptotic steady state at $x = 1$. The coordinate transformation $y = x - 1$ changes the equation into one having a unique steady state at $y = 0$, with $dy/dt = -y^3 - y^2 - y$. The coefficient, $c(x)$, can be taken as either a constant, $c = 1$, in which case $a(y) = -y^3 - y^2$, or with all the nonlinearity in the coefficient of y. In the latter case, $c(y) = y^2 + y + 1 > 0$, which reaches its absolute minimum at $y = -1/2$, a value unequal to the steady state, $y = 0$.*

Example 3.8 *The requirement that the system have a globally asymptotic steady state in order to be written in the desired form cannot be dropped. The equation $\dot{x} = be^{-cx}$, for b and c positive constants, is dissipative but has no steady state. It cannot be rewritten in the form of Eqn. (1).*

The formulation of an autonomous, first-order, differential equation in terms of coefficients, $c(x)$, and asymptotes, $a(x)$, is not unique. Some judgement and analysis as to which formulation leads to the most efficient numerical modeling is required.

4 Convergence of Nonlinear Algorithms

To be useful in numerically integrating evolution equations, integration algorithms must satisfy some basic properties. If the algorithm is implicit, the algorithms must be shown to be well defined. The algorithm must also produce the exact solution in the limit if the step sizes are taken very small. This convergence is usually a consequence of consistency and stability. An algorithm is stable if the approximation errors do not accumulate in such a way as to drive the approximate solution far from the exact solution after many steps. Consistency assures that an algorithm, which converges to some limit, actually converges to the exact solution. Finally, the algorithm should behave in the same way as the ordinary differential equation it is integrating. For example, if the differential equation has a globally asymptotic steady state, the algorithm should reproduce such behavior. It is in this property that the asymptotic integrators distinguish themselves from other integrators for dissipative differential equations.

For an algorithm to be well defined, it must be determined that the function $F(h, x_n) = x_{n+1}$ is an algorithm in the sense of ROSINGER (1980). When h is fixed, write $F_h(x_n) = F(h, x_n)$. Let T be the largest value that the time step, h, can take. A compact subset, K, of the real numbers is a finite union of closed intervals.

Definition 4.1 *An algorithm is a family of mappings*

$$F_h : X \to X \qquad \textit{with} \qquad h \in [0, T]$$

on a topological space, X, such that

1) F_o = identity on X

2) The function $[0,T] \times X \to X$ defined by $(h, x) \to F_h(x)$ is continuous in h and x.

3) For each compact set, $K \subset X$, there is a $M(K) > 0$, such that for any pair $u, v \in K$ and $h \in (0, T]$, the local Lipschitz condition holds:

$$|F_h(u) - F_h(v)| \leq [1 + M(K)h]|u - v|.$$

Furthermore, the stability and consistency of the algorithms must be investigated to verify that the algorithm converges to the exact solution. An algorithm is convergent if the global discretization error, $E = x(t + h) - x_{n+1}$ tends to zero as h tends to zero. In other words, an algorithm converges if, as the step size tends to zero, the approximate solution approaches the exact solution. Consistency is defined for an arbitrary algorithm involving n degrees of freedom, **X**, by MARSDEN (1992).

Definition 4.2 *An algorithm $F_h : \Re^n \to \Re^n$, for $\dot{\mathbf{X}} = f(\mathbf{X})$, such that $F_h(\mathbf{X}_n) = \mathbf{X}_{n+1}$ for $h > 0$ is called consistent or first order accurate if*

$$\left.\frac{dF_h(\mathbf{X}_n)}{dh}\right|_{h=0} = f(\mathbf{X}_n).$$

Higher order accuracy is defined similarly by requiring that the higher order derivatives of F_h are equal to the corresponding derivative of $x(t)$ as h tends to zero.

This definition agrees with the more traditional statement of consistency in which the algorithm is written in the form

$$x_{n+1} = x_n + h\phi(x_n, h)$$

and an algorithm is consistent if

$$\lim_{h \to 0} \phi(x_n, h) = \dot{x}(t_n).$$

The equivalence is seen by writing the Taylor series expansion for small h,

$$F_h(\mathbf{X}_n) = \mathbf{X}_n + \left.\frac{dF_h(\mathbf{X}_n)}{dh}\right|_{h=0} h + \ldots.$$

The final requirement is that the algorithm be stable. Stability for linear algorithms can be investigated by examining the position of the eigenvalues in the complex plane. Classically, the stability of methods such as the various Runge-Kutta methods have been studied by applying the method to the one-dimensional test equation, $\dot{x} = \lambda x$. A-stability, for example, is defined in terms of this linear test equation, see QUINNEY (1987, p. 77).

Definition 4.3 *A difference method is A-stable if when applied to the test problem, $dx/dt = \lambda x$, where $Re(\lambda) < 0$, all difference solutions, x_n, tend to zero as n tends to infinity.*

The asymptotic algorithms produce the exact solution to this equation and are therefore more stable than A-stable. It might be more important to test the nonlinear asymptotic algorithms on nonlinear, ordinary differential equations, such as a dissipative system.

Butcher proposed a definition of stability in 1975 to extend A-stability to Runge-Kutta methods applied to nonlinear, ordinary differential equations (HAIRER and WANNER 1991, p. 192). This is now called B-stability and can be applied to arbitrary methods, which may be nonlinear in the time step, h.

Definition 4.4 *A method is B-stable if when applied to contractive, ordinary differential equations $\dot{x} = f(t,x)$ with $\langle f(t,x) - f(t,y), x-y\rangle \leq 0$, then for all $h \geq 0$, $|x_{n+1} - y_{n+1}| \leq |x_n - y_n|$.*

B-stability implies A-stability. For example, the linear implicit (backward) Euler method is B-stable.

LAMBERT (1991) suggests that a useful definition for the stability of a nonlinear algorithm is that it be contractive, a generalization of B-stability to arbitrary algorithms. This condition reflects, in the algorithm, the fact that $|y(t) - x(t)|$ is a decreasing function of t if the differential equation is dissipative.

Definition 4.5 *If $\{x_n\}$ and $\{y_n\}$ are solutions generated from different initial conditions by an algorithm, then the algorithm is contractive if*

$$|x_{n+1} - y_{n+1}| \leq |x_n - y_n|.$$

Some authors, such as HUGHES (1983), have defined stability to mean that $|x_{n+1}| \leq |x_n|$. This condition is weaker than contractivity because it is implied by contractivity but does not imply contractivity.

Lemma 4.6 *Suppose an ordinary differential equation has a steady state $x^* = 0$. If the algorithm is contractive, then $|x_{n+1}| \leq |x_n|$.*

Proof: In $|x_{n+1} - y_{n+1}| \leq |x_n - y_n|$, let $y_n = y_{n+1} = 0$.

These ideas are generalized further in ROSINGER's (1980) definition of stability for nonlinear algorithms.

Definition 4.7 *A difference scheme is stable if for each compact set, $K \subset X$, there is a $L(K) > 0$, such that for any pair $u, v \in K, h \in (0,T], n \in N$ (where N is the set of positive integers), $nh \leq T$, the uniform local Lipschitz condition holds:*

$$|F_h^n(u) - F_h^n(v)| \leq L(K)|u - v|,$$

where F_h^n is the n^{th} iterate of F_h.

Definition 4.8 *The scheme is unconditionally stable if the algorithm is stable for $h \in (0,\infty)$ and conditionally stable if it is stable only for $h \in (0,T]$, where $T < \infty$.*

With these definitions, ROSINGER (1980) proved that for consistent algorithms, stability is equivalent to convergence. This is a generalization of the Lax-Richtmyer theorem for linear algorithms.

Theorem 4.9 *If an algorithm is first order accurate, then stability in the sense of Definition (4.7) is equivalent to convergence.*

5 Explicit Algorithm

The explicit asymptotic algorithm is clearly well defined. HENRICI (1962, p. 71) gives a theorem for the convergence of explicit methods in which a Lipschitz condition plays the role of stability.

Theorem 5.1 *Let the differential equation $dx/dt = f(t,x)$ have an approximate method defined by $x_{n+1} = x_n + h\phi(t, x_n, h)$, where $\phi(t,x,h)$ is continuous in the domain $D : t \in [a,b], x \in (-\infty, \infty)$, and $h \in [0, h_o]$, for $h_o > 0$. Suppose there is a constant L such that $|\phi(t,x,h) - \phi(t,y,h)| \leq L|x-y|$ for all (t,x,h) and (t,y,h) in D. Then the consistency condition $\phi(t,x,0) = f(t,x)$ is a necessary and sufficient condition for the convergence of the method defined by the increment function ϕ.*

This theorem can be applied to the explicit, one-step, asymptotic algorithm for the autonomous system, $\dot{x} = f(x) = c(x)[a(x) - x]$. Define $\phi(x,h)$ on the domain $D : t \in [a,b], x \in (-\infty, \infty)$, and $h \in [0, h_o]$, for some $h_o > 0$ by putting

$$\begin{aligned} \phi(x,h) &= \{[a(x) - x][1 - \exp(-c(x)h)]\}/h \qquad h \in (0, h_o] \\ \phi(x,0) &= f(x) \qquad h = 0. \end{aligned}$$

Lemma 5.2 *Suppose $dx/dt = f(x) = c(x)[a(x) - x]$, where $c(x)$ and $a(x)$ are continously differentiable for all x. Then $\phi(x,h)$ is continuous and continuously differentiable with respect to x on D.*

Proof: $\phi(x,h)$ is clearly continuous in x for $h > 0$. By L'Hospital's rule,

$$\lim_{h \to 0} \phi(x,h) = \lim_{h \to 0}\{[a(x) - x]c(x)e^{-c(x)h}\} = f(x) = \phi(x,0).$$

Therefore $\phi(x,h)$ is continuous in x, and h on D.

Since $a(x)$ and $c(x)$ are assumed to be continuously differentiable for all x, for all fixed $h \in (0, h_o]$, $d\phi/dx$ exists and is continuous. Define $d\phi/dx = df/dx$ at each $(x, 0)$. Then by L'Hospital's rule, for fixed x, at $h = 0$,

$$\begin{aligned} \lim_{h \to 0} d\phi/dx &= \lim_{h \to 0}\{[a(x) - x](dc/dx)he^{-c(x)h} + [da/dx - 1][1 - e^{-c(x)h}]\}/h \\ &= [a(x) - x](dc/dx) + [da/dx - 1]c(x) = df/dx. \end{aligned}$$

Theorem 5.3 *Suppose $dx/dt = f(x) = c(x)[a(x) - x]$, where $c(x)$ and $a(x)$ are continously differentiable for all x. The explicit algorithm $x_{n+1} = x_n + [a(x_n) - x_n][1 - \exp(-c(x_n)h)]$ is convergent if $|\phi(x,h) - \phi(y,h)| \leq L|x - y|$ for all (x,h) and (y,h) in D.*

Proof: The Henrici theorem gives the result immediately by the lemma, and because consistency holds, $f(x) = \phi(x, 0)$. The algorithm is convergent without any restriction on the sign of $c(x)$.

In general, the method is not unconditionally convergent since h_o must be chosen to make the Lipschitz condition in x hold for all (x,h) and (y,h) in D. This condition can be met if the derivative df/dx is bounded.

The continuous function, $\phi(x,h)$, satisfies the Lipschitz condition in x if $d\phi/dx$ is bounded for all $x \in (-\infty, \infty)$, by the mean value theorem. In this case, the algorithm is convergent. The goal then is to choose h_o so that

$$d\phi/dx = \{[a(x) - x](dc/dx)he^{-c(x)h} + [da/dx - 1][1 - e^{-c(x)h}]\}/h$$

is bounded for all $x \in (-\infty, \infty)$.

This condition for convergence is not always satisfied, even if a global solution exists for $dx/dt = f(x)$. For example, no such h_o exists for either of the systems

$$\begin{aligned} dx/dt &= c(bx^2 - x) \\ dx/dt &= bx(a - x), \qquad \text{for a, b, c constants.} \end{aligned}$$

The function, ϕ, associated with these differential equations does not satisfy the Lipschitz condition for all $x \in (-\infty, \infty)$.

5.1 Stability

DAHLQUIST (1963) has shown that no explicit, linear, multistep method is A-stable. However, the explicit asymptotic algorithm is an exact solution for the test equation, and therefore,

Lemma 5.4 *The explicit, one-step, asymptotic algorithm is A-stable.*

Proof: For the equation $dx/dt = \lambda x$, the algorithm becomes

$$x_{n+1} = x_n e^{\lambda h},$$

so that

$$|x_{n+1}| = |x_n|\,|e^{\lambda h}| \leq |x_n|\,|e^{Re(\lambda)h}|.$$

Therefore

$$|x_{n+1}| \leq |x_o|\,|e^{Re(\lambda)h}|^{n+1}.$$

But $|e^{Re(\lambda)h}| < 1$ for $Re(\lambda)h < 0$, and so $\lim_{n\to\infty} |x_n| = 0$. The method is A-stable.

The stability properties of the explicit asymptotic algorithm are therefore an improvement over any linear explicit method, such as the forward Euler. The explicit asymptotic algorithm is convergent when applied to nonlinear, ordinary differential equations satisfying a boundedness condition.

Theorem 5.5 *Suppose $dx/dt = f(x) = c(x)[a(x) - x]$, where $c(x) > 0$ and $a(x)$ are continuously differentiable for all x. The explicit algorithm $x_{n+1} = x_n + [a(x_n) - x_n][1 - \exp(-c(x_n)h)]$ is convergent if df/dx is bounded for all $x \in (-\infty, \infty)$. In this case, the algorithm is unconditionally stable.*

Proof: Since $[1 - e^{-c(x)h}]/h < c(x)$, and $e^{-c(x)h} < 1$, if $c(x) > 0$ and $h > 0$,

$$\begin{aligned} d\phi/dx &= \{[a(x) - x](dc/dx)he^{-c(x)h} + [(da/dx) - 1][1 - e^{-c(x)h}]\}/h \\ &< [a(x) - x](dc/dx) + [(da/dx) - 1]c(x) = df/dx. \end{aligned}$$

Therefore if df/dx is bounded for all $x \in (-\infty, \infty)$, so is $d\phi/dx$. Then by the mean value theorem, the Lipschitz condition holds for ϕ. By the Henrici theorem, the algorithm is convergent. It is unconditionally stable because there is no restriction on h.

Example 5.6 *The equation, $\dot{x} = -x^3 - x$, is a dissipative equation that is unbounded on the real numbers. If $c(x) = 1$ and $a(x) = -x^3$, the explicit algorithm is both unstable and oscillates around the globally asymptotic stable-state for a step size of $h = -\ln(0.5) = 0.693$ and an initial condition greater than 1. On the other hand, if all the nonlinearity is collected in the coefficient $c(x) = x^2 + 1$ so that $a(x) = 0$, the explicit algorithm is stable in the sense that $|x_{n+1}| \le |x_n|$ and does not oscillate about the steady state, for all h.*

In general, then, the algorithm is not stable for nonlinear, ordinary differential equations unless the step size is restricted. The stability of the explicit algorithm applied to a particular ordinary differential equation also depends on the formulation of that equation.

5.2 Error Estimate

The accuracy of the explicit asymptotic algorithm can be compared to the classical linear algorithm for small time steps. Estimates of the local truncation error can be obtained in the case that the ordinary differential equation $dx/dt = f(x)$ is written in the form

$$dx/dt = -c(x)\,x,$$

with $c(x) > 0$. Its exact solution is from Eqn. (2),

$$x(t+h) = x(t)\ \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta\right].$$

The explicit, one-step, asymptotic algorithm, for this case, is $x_{n+1} = x_n\ \exp[-c(x_n)h]$. Assume that $x_n = x(t)$.

The truncation error is the difference of the exact and approximate solutions,

$$E = x_{n+1} - x(t+h).$$

It is more effective to first compute the ratio,

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta + c(x_n)h\right]. \tag{7}$$

By the mean value theorem, for some $t < s < \eta$,

$$c(x(\eta)) - c(x_n) = (dc(s)/d\eta)\,(\eta - t).$$

Then

$$\begin{aligned}\left|\int_t^{t+h} c(x(\eta))\,d\eta - c(x_n)h\right| &= \left|\int_t^{t+h} [c(x(\eta)) - c(x_n)]\,d\eta\right| \\ &= \left|\int_t^{t+h} (dc(s)/d\eta)\,(\eta - t)\,d\eta\right| < Mh^2/2,\end{aligned}$$

where M is the maximum of $|dc/dt|$ on the closed interval $[t, t+h]$. Therefore, since $e^{-x} \ge e^{-|x|}$, the ratio is

$$x(t+h)/x_{n+1} \ge \exp(-Mh^2/2).$$

then

$$E \leq x_{n+1}[1 - \exp(-Mh^2/2)].$$

The local truncation error is proportional to h^2 for small h, and the cumulative error is proportional to h. This is the same error obtained for the classical forward Euler method.

The explicit algorithm was called underdamped by WALKER and FREED (1991) in its application to a preditor-prey model. The algorithmic response depends on whether the functions $c(x)$ and the exact solution $x(t)$ are increasing or decreasing over the time interval of interest. For example, by Eqn. (7), the ratio, of the exact solution and the approximation is

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h}[c(x(\eta)) - c(x_n)]d\eta\right].$$

If $x_n > 0$, $x(\eta)$ is decreasing and $c(x)$ is increasing, then the difference $c(x(\eta)) - c(x_n)$ is nonpositive for all η. This implies that $x(t+h)/x_{n+1} \geq 1$. In this case, therefore, the explicit method approximation x_{n+1} is less than $x(t+h)$.

Example 5.7 *The ordinary differential equation,* $\dot{x} = -x^3 - x$ *can be written as* $\dot{x} = (x^2+1)(-x)$ *with* $c(x) = x^2 + 1$. *For the initial condition* $x_n = 1$ *and* $h = 0.1$, *the exact solution is* $x(0.1) = 0.832523$, *and the explicit asymptotic algorithm produces* $x_{n+1} = 0.809675$.

6 Implicit Algorithms

The implicit solution, x_{n+1}, for any nonlinear implicit difference equation either may not exist or may not be unique. A numerical solution is usually sought at each step by either the Newton-Raphson method or by the method of successive approximations. Frequently, the Newton-Raphson method is the more efficient of the two methods. For implicit functions that have more than one solution, there is no guarantee to which root the Newton-Raphson method will converge, especially at larger time steps, so that it may not select the root closest to the exact solution to the differential equation.

As an example in which the backwards Euler algorithm is not well defined, consider $dx/dt = x^3 - x$. This equation has constant solutions $x = -1, x = 0$, and $x = 1$, which are the only possible horizontal asymptotes. The graphs of the families of solutions can be sketched by looking at the signs of the first derivative, $dx/dt = f(x) = x^3 - x$, and of the second derivative, $d^2f/dt^2 = (df/dx)(dx/dt) = (3x^2-1)(x^3-x)$. The concavity of the solutions changes whenever $x = \pm 1/\sqrt{3}$. Of the possible steady states, the only one that is an asymptote is $x = 0$ because $df/dx < 0$ there.

If $|x_n| < 1$, so that it is in the region where solutions are asymptotic to $x = 0$, then the backwards Euler method is not well defined for any step size $h > 0$. In this case, x_{n+1} is a solution of the equation $x_n + hx^3 - (1+h)x = 0$. The discriminant of the cubic is negative if $x_n^2 < 4(1+h)^3/27h$. The right hand side takes its minimum of 1 at $h = 1/2$. Therefore there are three unequal real solutions to the backwards Euler difference equation for all $h > 0$ and $|x_n| < 1$.

For $dx/dt = x^3 - x$, in the form $c[a(x) - x]$, so that $c = 1$ and $a(x) = x^3$, the asymptotic algorithm gives

$$x_{n+1} = x_n \exp(-h) + x_{n+1}^3[1 - \exp(-h)].$$

The discriminant for this cubic is negative if $x_n^2 < 4/[27\exp(-h)(1-\exp(-h))]$. The right hand side takes its minimum value of 1 when $h = \ln(3/2)$. Therefore again, if $|x_n| < 1$ there are three real and unequal solutions to the implicit asymptotic integrator difference equation for all $h > 0$.

One might try to repair this situation by letting $x_{n+1} = F(h, x_n)$ be the root chosen by the Newton-Raphson method. In other words, the algorithm would be a combination of the difference equation and the Newton-Raphson method. Unfortunately this function, $x_{n+1} = F(h, x_n)$, may not be continuous for fixed h and certainly will not always converge to an acceptable approximation of the solution to the differential equation. The Newton-Raphson method does not always choose the root closest to the initial point. For example, let $f(x) = x^3 - x = 0$. Then if one starts at 0.5, the method converges to the root -1. Furthermore, the Newton-Raphson method behaves very poorly with perturbations of the initial guess. If one makes an initial guess 0.4476, then the method converges to 1, if 0.4470 to 0, and if 0.4478 to -1. The Kantorovich theorem says only that Newton's method converges to a root within the radius of convergence; it does not say the root is the only or desired one in that set.

This problem can be avoided if the implicit algorithm has a unique solution. One condition which produces a unique solution is that the original differential equation have a globally asymptotic, stable steady-state, x^*. Assume that $x_n \neq x^*$, since if $x_n = x^*$, then the unique solution to the differential equation is known to be $x(t) = x^*$, and no numerical method is required. This result is also valid for the classical backward Euler method.

Lemma 6.1 *If $\dot{x} = f(x)$ is dissipative and if $f(x) = 0$ has a unique solution, then the backward Euler method has a unique solution for all step sizes h.*

Proof: Define the function $G(x) = x_n + hf(x) - x$. The backward Euler method has a unique solution if $G(x)$ has a unique root. $G(x)$ is monotonically decreasing because $dG/dx < 0$ if $df/dx < 0$. Assume that the unique steady state is $x^* = 0$. The unique root of $G(x)$ lies between x_n and x^* because $G(x_n) = hf(x_n)$ and $G(x^*) = x_n$ have opposite signs as a result of x^* being a globally asymptotic steady state. Therefore $G(x)$ has a unique root.

In the case of a dissipative, ordinary differential equation with a unique steady-state, the implicit asymptotic algorithm has the same property as the differential equation. It has a globally asymptotic stable steady-state as h tends to infinity. Since $\lim_{h\to\infty} x_{n+1} = a(\lim_{h\to\infty} x_{n+1})$, and because x^* is the unique value of x such that $x^* = a(x^*)$, it is also true that $\lim_{h\to\infty} x_{n+1} = x^*$.

6.1 Existence and Uniqueness

To verify that the asymptotic implicit algorithm is well defined for dissipative ordinary differential equations, it must be shown that x_{n+1} exists and is unique. Define the function

$$G(x) = x_n e^{-c(x)h} + a(x)(1 - e^{-c(x)h}) - x \tag{8}$$

for fixed x_n and h. The implicit difference equation has a unique solution for given x_n and h iff $G(x)$ has a unique root. The idea is to show that there is such a root between x_n and x^*.

Lemma 6.2 *If $dx/dt = c(x)[a(x) - x]$ with $c(x) \geq 0$ has a globally asymptotic equilibrium state, x^*, there is at least one root, r, of $G(x)$ between x^* and x_n.*

Proof: Either $c(x^*) = 0$ or $a(x^*) = x^*$ (or both). If $c(x^*) = 0$, then

$$G(x^*) = x_n - x^*.$$

If $a(x^*) = x^*$, then

$$G(x^*) = [x_n - x^*]e^{-c(x^*)h}.$$

Notice that the values of $G(x^*)$ agree if x^* is a root of both $c(x)$ and $a(x) - x$. Also,

$$G(x_n) = [a(x_n) - x_n][1 - e^{-c(x_n)h}].$$

If $x_n > x^*$, then for the curve $x(t)$ passing through x_n to be asymptotic to $x(t) = x^*$, dx/dt at x_n must be negative and $a(x_n) - x_n < 0$. Therefore $G(x^*)$ and $G(x_n)$ have opposite signs, and by the continuity of $G(x)$, a root must exist. A similar calculation produces the same result if $x_n < x^*$.

Any root of $G(x)$ must lie between x^* and x_n.

Lemma 6.3 *Suppose r is a root for $G(x)$ and $x_n \neq x^*$, then*

$$\begin{array}{ll} x^* < r < x_n & \textit{if} \quad x^* < x_n \\ x_n < r < x^* & \textit{if} \quad x^* > x_n \end{array}$$

Proof: The root $r \neq x^*$ since $G(x^*) \neq 0$. Because $G(r) = 0$, multiplying $G(r)$ by $c(r)$ and rearranging produces

$$c(r)(a(r) - r) = c(r)[a(r) - x_n]e^{-c(r)h}.$$

Therefore dx/dt evaluated at r can also be written as

$$dx/dt = c(r)[a(r) - x_n]e^{-c(r)h}.$$

If $x^* < r$, then because x^* is an asymptote and any solution must be decreasing, dx/dt evaluated at r is negative. The constraint $c(x) > 0$ implies that $a(r) - x_n < 0$. Since $G(x)$ can be rewritten as

$$G(x) = [a(x) - x_n](1 - e^{-c(x)h}) - (x - x_n),$$

$G(r) = 0$ implies that $[a(r) - x_n](1 - e^{-c(r)h}) = r - x_n$, so that $a(r) - x_n$ and $r - x_n$ have the same sign. Therefore $r < x_n$ and $x^* < r < x_n$. The second statement follows in a similar manner.

These two lemmas hold for x_n in any region in which the solutions are asymptotic to a constant solution both from above and from below. They will be useful in proving results that do not hold for all x_n and h.

The root can now be shown to be unique by showing that $G(x)$ is continuous and decreasing for all x near x^*. This result is more restrictive than necessary for a specific x_n because the monotonicity of $G(x)$ is not required outside of the region between x_n and x^*. But it is required to account for arbitrary x_n. The derivative of $G(x)$,

$$dG/dx = [a(x) - x_n](dc/dx)he^{-c(x)h} + (da/dx)(1 - e^{-c(x)h}) - 1 \qquad (9)$$

is negative when $h = 0$, so that for $G(x)$ to be monotonic, it must always be negative. The monotonicity of $G(x)$ for all h can be easily verified if $c(x)$ is a constant.

Theorem 6.4 *Let $dx/dt = f(x) = c[a(x) - x]$, where $c > 0$ is a constant, be such that $f(x) = 0$ defines a unique globally asymptotic steady-state at the origin. Then the one-step, implicit, asymptotic algorithm has a unique solution for all step sizes h and all initial conditions.*

Proof: Because the origin is a globally asymptotic steady-state, $df/dx < 0$ for all x and consequently $da/dx - 1 < 0$. Since $da/dx - 1 < 0$ and since $0 < 1 - e^{-ch} < 1$, then

$$dG/dx = (da/dx)(1 - e^{-ch}) - 1 < 0.$$

Therefore the function $G(x)$ is monotonic decreasing and has a unique root.

If $c(x)$ is not constant, then the solution may not be unique for all step sizes h. The behavior depends on the sign and magnitude of dc/dx. The following theorem covers the case when the coefficient $c(x)$ is increasing for $x > x^*$ and decreasing otherwise.

Theorem 6.5 *Suppose that the ordinary differential equation $dx/dt = c(x)(a(x) - x)$ with $c(x) \geq 0, da/dx \leq 0$ has a globally asymptotic, stable steady-state, x^*, and that dc/dx and $a(x) - x_n$ have opposite signs when x is between x^* and x_n. Then the one-step, implicit, asymptotic algorithm has a unique solution for all time steps, h.*

Proof: This follows from the two previous Lemmas and the monotonicity of $G(x)$.

Example 6.6 *The implicit algorithm applied to the differential equation, $\dot{x} = -x^3 - x$ written as $\dot{x} = -(x^2 + 1)x$, with $c(x) = x^2 + 1$ and $a(x) = 0$, has a unique solution due to this theorem.*

The behavior in general can be visualized from the fact that the term $he^{-c(x)h}$ varies from 0 when $h = 0$ to a maximum of $e^{-1}/c(x)$ at $h = 1/c(x)$ and then asymptotically back to zero as h tends to infinity, for fixed x. Because $(da/dx)(1 - e^{-c(x)h}) - 1$ is negative for all h, in the case that $[a(x) - x_n](dc/dx)$ is positive, there is a range of h near zero, and another range of large h, such that $dG/dx < 0$, so that the implicit algorithm has a unique solution for those h.

6.2 Existence and Uniqueness by Contractions

The use of the Newton-Raphson method to solve implicit nonlinear algorithms can lead to numerical difficulties. If the solution to the implicit algorithm can be shown to be the fixed point of a contractive function, then the method of successive approximations can often be used more efficiently in place of the Newton-Raphson method. The idea of a contractive mapping from a metric space to itself produces conditions under which an implicit algorithm can be shown to have a unique solution.

Definition 6.7 *A function $f : M \rightarrow M$, where M is a complete metric space, is a contraction if*

$$|f(x) - f(y)| \leq k|x - y| \quad \textit{for} \quad 0 \leq k < 1$$

for all $x, y \in M$.

A contraction has a unique fixed point, and the fixed point can be found by successive approximations starting at any initial point.

An implicit algorithm is defined by the relation $x_{n+1} = g(x_{n+1}, x_n)$. For a given x_n, the solution x_{n+1} is a fixed point of the function $g(x)$. If the original differential equation has a globally asymptotic steady-state, then x_{n+1} must lie between x_n and 0. Think of $g(x)$ defined on the complete metric space $[0, x_n]$. To prove that $g(x)$ is a contraction on $[0, x_n]$, it is sufficient to prove that the absolute value of dg/dx on $[0, x_n]$ is less than one, since the mean value theorem says that $|g(x) - g(y)| = \left|\frac{dg}{dx}(z)\right| |x - y|$ for some $x < z < y$.

For example, if all the nonlinearity is in the coefficient, $c(x)$, then the implicit, one-step, asymptotic algorithm is $x_{n+1} = x_n \exp[-c(x_{n+1})h]$. In this case, $g(x) = x_n \exp[-c(x)h]$. It is then possible to restrict the time step, h, so that $g(x)$ is a contraction. Let c^* be the maximum of $|dc/dx|$ on the interval $[0, x_n]$; c^* exists because the interval is closed.

$$|dg/dc| = |x_n(dc/dx)h \exp[-c(x)h]| \leq |x_n(dc/dx)h| \leq |x_n c^* h| < 1$$

Therefore the function $g(x)$ is contractive on the interval if

$$h < 1/(x_n c^*).$$

The more slowly that $c(x)$ changes, the larger the step size possible. Under this restriction on h, the implicit algorithm has a unique solution, x_{n+1}, the fixed point of $g(x)$ and this point can be computed by successive approximations starting at, say, x_n. This technique may be more efficient than the Newton-Raphson method, plus it guarantees that the solution obtained is unique. Notice further, that as the computed value of the solution gets closer to the steady state, 0, the size of the allowable step size increases.

6.3 Stability

Conditions have been found for the implicit difference equation to have a unique root, x_{n+1}, so that the algorithmic function $F(h, x_n) = x_{n+1}$ is well defined. Recall that $F_h(x_n) \equiv F(h, x_n) = x_{n+1}$. To investigate the stability, one needs to know how the algorithm applied to dissipative differential equations behaves if the initial condition, x_n, is perturbed to an initial condition, y_n. The following discussion depends on the fact that x_n can be written explicitly in terms of x_{n+1} for fixed h as $F_h^{-1}(x_{n+1}) = x_n$, where, writing x for x_{n+1},

$$F_h^{-1}(x) = xe^{c(x)h} - a(x)(e^{c(x)h} - 1).$$

Then

$$\begin{aligned} dF_h^{-1}/dx &= e^{c(x)h} + x(dc/dx)he^{c(x)h} \\ &\quad -(da/dx)(e^{c(x)h} - 1) - a(x)(dc/dx)he^{c(x)h} \\ &= e^{c(x)h} - (da/dx)(e^{c(x)h} - 1) + [x - a(x)](dc/dx)he^{c(x)h}. \end{aligned}$$

The following lemma will be useful in the investigation of stability.

Lemma 6.8 *Let $dx/dt = f(x) = c(x)[a(x) - x]$, where $c(x) > 0$ and $a(x)$ is monotonically decreasing, be such that $f(x) = 0$ has a unique globally asymptotic steady-state at the origin. Suppose either a) $c(x)$ is a constant or b) dc/dx and $a(x) - x_n$ have opposite signs when x is between x^* and x_n. The unique root, x_{n+1} is an increasing function of x_n for fixed h.*

Proof: In case a) $dc/dx = 0$ implies that $dF_h^{-1}/dx > 0$. In case b), because $da/dx < 0$ and $[x - a(x)](dc/dx) > 0, dF_h^{-1}/dx > 0$. Therefore $F_h^{-1}(x)$ is increasing and has an increasing inverse. This implies that $x_{n+1} = F_h(x_n)$ is an increasing function of x_n.

Lemma 6.9 *Suppose $a(x)$ is monotonically decreasing and either a) $c(x)$ is a constant or b) dc/dx and $a(x) - x_n$ have opposite signs when x is between x^* and x_n. Then the algorithm is contractive, $|x_n - y_n| > |x_{n+1} - y_{n+1}|$, for all h.*

Proof: Suppose that $x_n > y_n$. Put $x = x_{n+1}$ and $y = y_{n+1}$. Recall that $x_n = [x-a(x)]\{\exp[c(x)h]-1\} + x$. Then

$$x_n - y_n = [x - a(x)]\{\exp[c(x)h] - 1\} - [y - a(y)]\{\exp[c(y)h] - 1\} + x - y.$$

Verify that the function $H(x) = [x - a(x)]\{\exp[c(x)h] - 1\}$ is increasing by showing that

$$dH/dx = [x - a(x)](dc/dx)h \exp[c(x)h] + [1 - (da/dx)]\{\exp[c(x)h] - 1\} > 0.$$

In case a), $da/dx < 0$ implies that $dH/dx = [1 - (da/dx)]\{\exp[c(x)h] - 1\} > 0$. In case b), $da/dx < 0$ and $[x - a(x)](dc/dx) > 0$ imply that $dH/dx > 0$. Now $x_n > y_n$ implies that $x > y$ since x_{n+1} is an increasing function of x_n by the previous Lemma. $H(x)$ increasing gives

$$[x - a(x)]\{\exp[c(x)h] - 1\} - [y - a(y)]\{\exp[c(y)h] - 1\} > 0.$$

Therefore, $x_n - y_n > x_{n+1} - y_{n+1} > 0$ and $|x_n - y_n| > |x_{n+1} - y_{n+1}|$, for all h.

Any contractive algorithm satisfies ROSINGER's local Lipschitz condition.

Theorem 6.10 *Under the conditions that the implicit algorithm is contractive, it is also convergent.*

Proof: The result follows from ROSINGER's theorem (4.9). The implicit algorithm is first-order accurate, or consistent, for any ordinary differential equation in the form $\dot{x} = c(x)[a(x) - x]$. The derivative of x_{n+1} with respect to h is

$$\begin{aligned} dx_{n+1}/dh &= [-(dc/dx_{n+1})(dx_{n+1}/dh)h - c(x_{n+1})]\{x_n \exp[-c(x_{n+1})h] - a(x_{n+1})\} \\ &+ \{1 - \exp[-c(x_{n+1})h]\}(da/dx_{n+1})(dx_{n+1}/h). \end{aligned}$$

Therefore, the algorithm is first-order accurate because

$$\lim_{h \to o} dx_{n+1}/dh = c(x_n)[a(x_n) - x_n] = \dot{x}(x_n).$$

Consequently, the algorithm is convergent whenever is it unconditionally stable.

6.4 Error Estimate

Estimates of the local truncation error for small step size h can be obtained in the case that the ordinary differential equation $dx/dt = f(x)$ is written in the form

$$dx/dt = -c(x)\,x,$$

with $c(x) > 0$. Its exact solution is

$$x(t+h) = x(t)\ \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta\right].$$

The implicit, one-step, asymptotic algorithm, for this case, is $x_{n+1} = x_n\ \exp[-c(x_{n+1})h]$. Assume that $x_n = x(t)$. The ratio of the exact and approximate solutions is

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta + c(x_{n+1})h\right].$$

The local truncation error, E, is defined by $x(t+h) + E = x_{n+1}$. Since $c(x)$ is continuous and differentiable, there is a bound L such that

$$|c(x(t+h)) - c(x_{n+1})| \leq L|x(t+h) - x_{n+1}| = L|E|.$$

Then

$$\begin{aligned} \left|\int_t^{t+h} c(x(\eta))\,d\eta - c(x_{n+1})h\right| &= \left|\int_t^{t+h} [c(x(\eta)) - c(x(t+h)) + c(x(t+h)) - c(x_{n+1})]\,d\eta\right| \\ \leq \left|\int_t^{t+h} (dc(s)/d\eta)\,(t+h-\eta)\,d\eta\right| &+ \left|\int_t^{t+h} [c(x(t+h)) - c(x_{n+1})]\,d\eta\right| \\ &< Mh^2/2 + L|E|h, \end{aligned}$$

where M is the maximum of $|dc/dt|$ on the closed interval $[t, t+h]$. Therefore, the ratio is

$$x(t+h)/x_{n+1} \geq \exp(-Mh^2/2 - L|E|h).$$

Since $E = x_{n+1}(1 - x(t+h)/x_{n+1}$,

$$E \leq x_{n+1}\,[1 - \exp(-Mh^2/2 - L|E|h)].$$

E is proportional to $x_{n+1}(Mh^2/2 + L|E|h)$. Therefore, the local truncation error is proportional to h^2 for small h, and the cumulative error is proportional to h. This is the same error obtained for the classical backward Euler method and for the first-order, forward, asymptotic algorithm.

The implicit, asymptotic algorithm was said to produce overdamping by Walker and Freed (1991).

Lemma 6.11 *If $\dot{x} = -c(x)x$ and if $c(x) > 0$ is increasing, then whenever $x_n > 0$, $x_{n+1} > x(t+h)$.*

Proof: Assume the contrary, that $x_{n+1} < x(t+h)$. Since $x_n > 0$, the exact solution is decreasing. In the integral,

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h} [c(x(\eta)) - c(x_{n+1})]\,d\eta\right],$$

the term

$$c(x(\eta)) - c(x_{n+1}) = [c(x(\eta)) - c(x(t+h))] + [c(x(t+h)) - c(x_{n+1})]$$

is positive since $c(x)$ is increasing. Therefore

$$x(t+h)/x_{n+1} < 1 \qquad \text{and} \qquad x(t+h) < x_{n+1},$$

which is a contradiction.

Example 6.12 *The ordinary differential equation, $\dot{x} = -x^3 - x$ can be written as $\dot{x} = (x^2 + 1)(-x)$ with $c(x) = x^2 + 1$. For the initial condition $x_n = 1$ and $h = 0.1$, the exact solution is $x(0.1) = 0.832523$, and the implicit asymptotic algorithm produces $x_{n+1} = 0.842796$.*

7 Two-Step Midpoint Algorithm

The classical, linear, midpoint method approximates the integral of $\dot{x} = f(x)$ from x_n to x_{n+1} by a rectangle having base h and height taken at the midpoint of the interval $[t, t+h]$. HUGHES (1983, p. 145), generalized this to a nonlinear algorithm, in terms of x_n and x_{n+1}, by taking the height at $x_{n+\alpha} = (1-\alpha)x_n + \alpha x_{n+1}$. In other words, $x_{n+\alpha}$ is a linear interpolation between x_n and x_{n+1}. The nonlinear midpoint algorithm of HUGHES (1983, p. 144) and of HOGGE (1978) was applied to equations of the form, $M(x,t)(dx/dt) + K(x,t)x = F(t)$, by substituting $x_{n+\alpha}$ to get $M_{n+\alpha}v_{n+\alpha} + K_{n+\alpha}x_{n+\alpha} = F_{n+\alpha}$, where $v = dx/dt$. Because $x_{n+\alpha}$ is a linear function of x_n and x_{n+1}, the expressions in terms of $n+\alpha$ drop out, and one solves for x_{n+1}.

The asymptotic midpoint algorithm of Eqn. (5) can be viewed as a combination of the results of one single-step algorithm on part of the interval followed by another single-step algorithm on the remainder of the interval. If one uses the result from the implicit integrator, Eqn. (4), taken over the interval $[t, t+\theta h]$ as the initial condition for the explicit integrator, Eqn. (3), taken over the remaining interval $[t+\theta h, t+h]$, one obtains an algorithm in which both x_{n+1} and the θ-midpoint, $x_{n+\theta}$, must be determined. This is a two-step algorithm. The midpoint algorithm can be expressed as

$$x_{n+1} = x_n \exp(-c(x_{n+\theta})h) + [1 - \exp(-c(x_{n+\theta})h)]a(x_{n+\theta}),$$

in agreement with Eqn. (5), where, for $c_{n+\theta} = c(x_{n+\theta})$ and $a_{n+\theta} = a(x_{n+\theta})$,

$$x_{n+\theta} = x_n \exp(-\theta c_{n+\theta}h) + [1 - \exp(-\theta c_{n+\theta}h)]a_{n+\theta}.$$

The value $x_{n+\theta}$ exists under the same restrictions applying to the implicit asymptotic algorithm; it is obtained from the implicit method applied over the interval $[t, t+\theta h]$. In those cases, x_{n+1} exists since it is defined explicitly from $x_{n+\theta}$. For example, if $c(x)$ is a constant, then

$$x_{n+1} = x_n \exp(-ch) + [1 - \exp(-ch)]a(x_{n+\theta}),$$

and the midpoint algorithm is well defined for all $0 \leq \theta \leq 1$.

The midpoint algorithm, when $\theta = 1/2$, is more accurate than any of the other two-step θ-midpoint algorithms since, for small h, it approximates the second derivative of the solution as well as the first derivative.

Lemma 7.1 *The two-step midpoint algorithm is first-order accurate for all* $0 \leq \theta \leq 1$. *It is second-order accurate iff* $\theta = 1/2$.

Proof: The derivative of x_{n+1} with respect to h is

$$dx_{n+1}/dh = [(dc_{n+\theta}/dx)\dot{x}\theta h - c_{n+\theta}]\exp(-c_{n+\theta}h)(x_n - a_{n+\theta}) + [1 - \exp(-c_{n+\theta}h)](da_{n+\theta}/dx)\dot{x}\theta.$$

Letting h tend to zero,

$$\left.\frac{dx_{n+1}}{dh}\right|_{h=0} = -c_n x_n + c_n a_n = \dot{x}_n.$$

Therefore the algorithm is first-order accurate for all $0 \leq \theta \leq 1$. Likewise, taking second derivatives with respect to h shows that

$$\left.\frac{d^2 x_{n+1}}{dh^2}\right|_{h=0} = 2\theta(dc_n/dx)(a_n - x_n)\dot{x}_n + 2\theta c_n(da_n/dx)\dot{x}_n - c_n\dot{x}_n$$

But

$$\ddot{x} = (df/dx)\dot{x} = (dc_n/dx)(a_n - x_n)\dot{x}_n + c_n(da_n/dx)\dot{x}_n - c_n\dot{x}_n,$$

where $\dot{x} = f(x)$. Therefore

$$\left.\frac{d^2x_{n+1}}{dh^2}\right|_{h=0} = \ddot{x}_n \quad \text{iff} \quad \theta = 1/2.$$

Only when $\theta = 1/2$ is the midpoint algorithm second-order accurate.

7.1 Stability

This midpoint asymptotic algorithm reduces to the trapezoidal algorithm defined by FREED *et al.* (1992) if the function $a(x)$ is identically zero. This is the form of the ordinary differential equation that HUGHES (1983) took as the test case. Hughes tests the stability of his algorithm by application to the model equation,

$$dx/dt + c(x,t)x = 0 \quad \text{for} \quad c(x,t) > 0.$$

If this system is linear, his trapezoidal and midpoint algorithms agree. Stability is defined by Hughes as requiring $|x_{n+1}| < |x_n|$. His midpoint algorithm is unconditionally stable if $\alpha \geq 1/2$, and if $\alpha < 1/2$, it is stable whenever h satisfies

$$c_{n+1}h \leq 2/(1 - 2\alpha).$$

As Hughes points out, this shows that while the trapezoidal algorithm is unconditionally stable when applied to linear problems, it is not when applied to nonlinear problems. The asymptotic midpoint algorithm, on the contrary, is unconditionally stable for the same values of α when applied to both linear equations and to the nonlinear family with all nonlinearity in the coefficient of x.

If the asymptotic midpoint algorithm is applied to Hughes' model equation $dx/dt + c(x)x = 0$ for $c(x) > 0$, which has a unique steady-state $x = 0$, then

$$x_{n+1} = x_n \exp[-c(x_{n+\theta})h].$$

No matter what $c(x_{n+\theta})$ is, $|x_{n+1}| < |x_n|$ for all h and θ. Therefore, in this weaker definition of stability, the asymptotic midpoint algorithm, in the case where the asymptote is zero, is unconditionally stable for all $0 \leq \theta \leq 1$, at least under the Hughes definition of stability. As such, it is a more stable algorithm than the Hughes, nonlinear, midpoint algorithm.

7.2 Local Truncation Error

For small h, the linear midpoint algorithm has a local truncation error of order h^3, while the forward Euler method has a truncation error proportional to h^2 for small h. A similar improvement is provided by the asymptotic midpoint algorithm.

Lemma 7.2 *The local truncation error for the midpoint algorithm applied to $\dot{x} = -c(x)x$ is proportional to h^3 iff $\theta = 1/2$. Otherwise, it is proportional to h^2.*

Proof: Assume that $x(t) = x_n$. The ratio of the exact to the approximate solution over the interval $[t, t+h]$ is

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta + c(x_{n+\theta})h\right].$$

Take the Taylor series of $c(x(\eta))$ about $t_\theta = t + \theta h$,

$$c(\eta) = c(t_\theta) + (\eta - t_\theta)(dc(t_\theta)/dt) + [(\eta - t_\theta)^2/2](d^2c(\xi)/dt^2),$$

for some ξ such that $t_\theta \le \xi \le \eta$. The local truncation error requires the computation of

$$\begin{aligned} T_n &= \int_t^{t+h} c(\eta)d\eta - c(x_{n+\theta})h \\ &= c(t_\theta)h - c(x_{n+\theta})h + \int_t^{t+h}(\eta - t_\theta)(dc(t_\theta)/dt)d\eta + \int_t^{t+h}[(\eta - t_\theta)^2/2](d^2c(\xi)/dt^2)d\eta \\ &= c(t_\theta)h - c(x_{n+\theta})h + [(1-\theta)^2/2 - \theta^2/2]h^2(dc(t_\theta)/dt) + O(h^3). \end{aligned}$$

Since $c(x)$ is continuous and differentiable, there is an L such that

$$c(x_{n+\theta}) - c(x(t_\theta)) = L[x_{n+\theta} - x(t_\theta)].$$

Therefore,

$$T_n = LE_\theta h + [(1-\theta)^2/2 - \theta^2/2]h^2c'(t_\theta) + O(h^3),$$

where E_θ is the order h^2, local error from the asymptotic, implicit algorithm. T_n is of order h^3 if and only if the coefficient of h^2 is zero.

$$(1-\theta)^2/2 - \theta^2/2 = 0 \qquad \text{iff} \qquad \theta = 1/2.$$

Since the local truncation error is $E \le x_{n+1}[1 - \exp(-|T_n|)]$, E is proportional to h^3 iff $\theta = 1/2$.

This agrees with the previous results that the truncation error for both the asymptotic explicit ($\theta = 0$) and implicit ($\theta = 1$) methods is of order h^2.

8 One-Step Midpoint Algorithm

A one-step, generalized, midpoint method is defined in which first the explicit asymptotic algorithm and then the implicit method is applied. The order of application of the explicit and the implicit algorithms is the opposite of that employed to define the two-step, generalized, midpoint algorithm. The explicit asymptotic integrator, Eqn. (3), taken over the interval $[t, t+\, (1-\theta)h]$ gives the initial condition for the implicit asymptotic integrator, Eqn. (4), taken taken over the remaining interval $[t+\, (1-\theta)h, t+h]$, The form of the algorithm is

$$x_{n+1} = x_n + (1 - \exp\{-[(1-\theta)c_n + \theta c_{n+1}]h\})\,(a_n - x_n) + (1 - \exp[-\theta c_{n+1}h])\,(a_{n+1} - a_n). \quad (10)$$

When $\theta = 0$, the explicit asymptotic algorithm is recovered, and when $\theta = 1$, the implicit asymptotic algorithm is recovered.

If the time coefficient, c, is a constant, and the asymptote is monotonically decreasing, the one-step, asymptotic, midpoint algorithm is uniquely defined. If c is constant, the one-step midpoint algorithm can be written as

$$\begin{aligned} x_{n+1} &= (x_n - a_n)\exp(-ch) + a_n \exp(-\theta ch) \\ &+ [1 - \exp(-\theta ch)]a_{n+1}. \end{aligned}$$

If in addition a is a constant so that the original equation is linear,

$$x_{n+1} = [x_n - a]e^{-ch} + a.$$

The one-step midpoint algorithm gives the exact solution for linear equations, just as the other asymptotic integrators do.

Theorem 8.1 *If the dissipative equation $\dot{x} = c[a(x) - x]$ has $c > 0$ constant and $da/dx < 0$, then the one-step midpoint algorithm is well defined.*

Proof: Define the auxiliary function,

$$\begin{aligned} G(x) &= (x_n - a_n)\exp(-ch) + a_n \exp(-\theta ch) \\ &+ [1 - \exp(-\theta ch)]a(x) - x. \end{aligned}$$

The derivative, $dG/dx = [1 - \exp(-\theta ch)](da/dx) - 1 < 0$ for all h and θ so that G is monotonic in x for any fixed h. If $G(x)$ crosses the x axis, this root is unique.

$$G(x_n) = [1 - \exp(-ch)](a_n - x_n).$$

$$G(x^*) = (x_n - a_n)\exp(-ch) + [a_n - a(x^*)]\exp(-\theta ch),$$

since $a(x^*) = x^*$. Because x^* is a globally asymptotic steady-state and because $a(x)$ is decreasing, the sign of the terms, $(x_n - a_n)$ and $[a_n - a(x^*)]$ are the same and opposite to $(a_n - x_n)$. Therefore $G(x_n)$ and $G(x^*)$ have opposite signs, and the unique x_{n+1} lies between x_n and x^*.

If the function $a(x)$ is identically zero, this midpoint algorithm also reduces to the trapezoidal algorithm of FREED *et al.* (1992). In this case,

$$x_{n+1} = x_n \exp[-(1 - \theta)c_n h - \theta c_{n+1} h]. \tag{11}$$

Theorem 8.2 *The one-step midpoint algorithm applied to the system $\dot{x} = -c(x)x$, with $c(x) > 0$, is well-defined for all $h > 0$ and θ.*

Proof: Put

$$G(x) = x_n \exp[-(1 - \theta)c_n h]\exp[-\theta c(x)h] - x.$$

Then for $x^* = 0$,

$$\begin{aligned} G(x_n) &= -x_n(1 - \exp[-c(x_n)h]) \\ G(x^*) &= x_n \exp[-(1 - \theta)c_n h]\exp[-\theta c(0)h] \end{aligned}$$

have opposite signs. Therefore if $G(x)$ is monotonic between x_n and 0, it must have a unique root and the algorithm is well defined. This calculation depends on the facts that since the differential equation has a globally asymptotic steady-state,

$$-c(x) - (dc/dx)x < 0;$$

and

$$\theta h \exp[-\theta c(x)h] < 1/c(x).$$

If $x_n > x > 0$, then

$$\begin{aligned} dG/dx &= x_n \exp[-(1-\theta)c(x_n)h][-(dc/dx)\theta h] \exp[-\theta c(x)h] - 1 \\ &< [c(x)\theta h] \exp[-(1-\theta)c(x_n)h] \exp[-\theta c(x)h] - 1 \\ &< \exp[-(1-\theta)c(x_n)h] - 1 < 0. \end{aligned}$$

Therefore there is a unique value x_{n+1} satisfying the asymptotic, one-step, midpoint algorithm for all h and θ.

The accuracy of this midpoint algorithm is also improved when $\theta = 1/2$ since, for small h, it approximates the second derivative of the solution as well as the first derivative.

Lemma 8.3 *The one-step midpoint algorithm is first-order accurate for all* $0 \leq \theta \leq 1$. *It is second-order accurate iff* $\theta = 1/2$.

Proof: The derivative of x_{n+1} with respect to h is, by Eqn. (10),

$$\begin{aligned} \frac{dx_{n+1}}{dh} &= [(1-\theta)c_n + \theta c_{n+1} + \theta(dc/dx)(dx_{n+1}/dh)h]\,(1 - \exp\{-[(1-\theta)c_n + \theta c_{n+1}]h\})\,(a_n - x_n) \\ &+ [\theta c_{n+1} + \theta(dc/dx)(dx_{n+1}/dh)h] \exp[-\theta c_{n+1}h](a_{n+1} - a_n) \\ &+ (1 - \exp[-\theta c_{n+1}h])\,(da/dx)(dx_{n+1}/dh) \end{aligned}$$

Letting h tend to zero,

$$\left.\frac{dx_{n+1}}{dh}\right|_{h=0} = c_n(a_n - x_n) = \dot{x}_n.$$

Therefore the algorithm is first order accurate for all $0 \leq \theta \leq 1$. Likewise, taking second derivatives with respect to h shows that for arbitrary θ,

$$\left.\frac{d^2x_{n+1}}{dh^2}\right|_{h=0} = 2\theta(dc_n/dx)(a_n - x_n)\dot{x}_n + 2\theta c_n(da_n/dx)\dot{x}_n - c_n\dot{x}_n$$

But

$$\ddot{x} = (df/dx)\dot{x} = (dc_n/dx)(a_n - x_n)\dot{x}_n + c_n(da_n/dx)\dot{x}_n - c_n\dot{x}_n,$$

where $\dot{x} = f(x)$. Therefore

$$\left.\frac{d^2x_{n+1}}{dh^2}\right|_{h=0} = \ddot{x}_n \text{ iff } \theta = 1/2.$$

When $\theta = 1/2$, the one-step midpoint algorithm is second-order accurate.

8.1 Stability

If the asymptotic, one-step, midpoint algorithm is applied to $dx/dt + c(x)x = 0$ for $c(x) > 0$, which has unique steady-state $x = 0$, then x_{n+1} is given by Eqn. (11). If one puts $c_{n+\theta} = (1-\theta)c(x_n) + \theta c(x_{n+1})$, then

$$x_{n+1} = x_n \exp[-c_{n+\theta}h]$$

In any case, $|x_{n+1}| < |x_n|$ for all h and θ. Therefore, like the two-step method, the asymptotic, one-step, midpoint algorithm, in the case that the asymptote is zero, is unconditionally stable in this weak sense for all $0 \le \phi \le 1$.

8.2 Local Truncation Error

The local truncation error, for small h, improves to order h^3 for the one-step, asymptotic, midpoint algorithm iff $\theta = 1/2$.

Lemma 8.4 *The local truncation error for the one-step, asymptotic, midpoint algorithm applied to $\dot{x} = -c(x)x$ is proportional to h^3 iff $\theta = 1/2$. Otherwise, it is proportional to h^2.*

Proof: Assume that $x(t) = x_n$. The ratio of the exact to the approximate solution over the interval $[t, t+h]$ is

$$x(t+h)/x_{n+1} = \exp\left[-\int_t^{t+h} c(x(\eta))\,d\eta + (1-\theta)c(x_n)h + \theta c(x_{n+1})h\right].$$

The integral in the exponent can be expanded using Taylor series.

$$\begin{aligned}
T_n = & \int_t^{t+h} c(x(\eta))\,d\eta + (1-\theta)c(x_n)h + \theta c(x_{n+1})h \\
= & \int_t^{t+(1-\theta)h} [c(x(\eta)) - c_n]\,d\eta + \int_{t+(1-\theta)h}^{t+h} [c(x(\eta)) - \theta c_{n+1}]\,d\eta \\
= & \frac{dc(t)}{dt}\int_t^{t+(1-\theta)h} [(\eta - t) + O(2)]\,d\eta + \frac{dc(t+h)}{dt}\int_{t+(1-\theta)h}^{t+h} [(\eta - t - h) + O(2)]\,d\eta \\
= & \frac{dc(t)}{dt}(1-\theta)^2h^2/2 - \frac{dc(t+h)}{dt}\theta^2h^2/2 + O(h^3) \\
= & \frac{dc(t)}{dt}[(1-\theta)^2h^2/2 - \theta^2h^2/2] + \left[\frac{dc(t)}{dt} - \frac{dc(t+h)}{dt}\right]\theta^2h^2/2 + O(h^3) \\
= & \frac{dc(t)}{dt}[(1-\theta)^2h^2/2 - \theta^2h^2/2] + O(h^3),
\end{aligned}$$

since $dc(t)/dt$ is continuous and differentiable implies that $dc(t)/dt - dc(t+h)/dt$ is proportional to h. Therefore, the integral is proportional to h^3 iff $\theta = 1/2$.

Since the local truncation error is $E \le x_{n+1}[1 - \exp(-|T_n|)]$, E is proportional to h^3 iff $\theta = 1/2$.

This also agrees with the previous results that the truncation error for both the asymptotic explicit ($\theta = 0$) and implicit ($\theta = 1$) methods is of order h^2.

9 Summary

Three nonlinear asymptotic algorithms proposed by WALKER and FREED (1991) have been investigated for one-dimensional, dissipative, ordinary differential equations of the form, $\dot{x} = c(x)[a(x) - x]$ for $c(x) > 0$ with a globally asymptotically stable-steady state. A new one-step, midpoint, asymptotic algorithm has been proposed.

A method related to those studied here was developed by DENNIS (1959) to integrate homogeneous, first-order, ordinary differential equations with exponential type solutions. In the equation $\dot{x} = f(x)$, the substitution $u = ln(x)$ transforms the ordinary differential equation to $\dot{u} = f/x$. The algorithm $x_{r+p} = x_r \exp[\int g\,dt]$, where $g = f/x$, is obtained in which a numerical solution is taken for the integral. If the exact solution is exponential, then u varies slowly, and the algorithm gives good results.

To apply these asymptotic algorithms, the ordinary differential equation must be written in the form, $\dot{x} = c(x)[a(x)-x]$. Such a formulation is not unique, and the choice affects the behavior of the algorithms. If the equation is dissipative and has a globally asymptotic steady-state, then the equation can always be written in the form, $\dot{x} = -c(x)x$ with $c(x) > 0$.

The algorithms are particularly studied for two extreme cases of dissipative differential equations with a globally asymptotic steady-state: that $c(x)$ is constant or that $a(x) = 0$. The differential equation with $c(x)$ a constant is a generalization of the semi-linear equations. The implicit algorithm is well-defined for any step size h if $c(x)$ is constant. In the second case, a bound on h was found for the implicit integrator to have a unique solution and, in fact, to be contractive. In the first case, the implicit algorithm is contractive for all $h > 0$. They may be efficiently solved with the method of successive approximations.

Each of the asymptotic algorithms produces the exact solution for any linear differential equation and are, therefore, more successful than the classical linear methods. In the case that $a(x) \equiv 0$, both midpoint integrators, and the trapezoidal integrator proposed in FREED *et al.* (1992), reduce to the identical algorithm.

All of the algorithms are first-order accurate. Both midpoint methods are second-order accurate iff $\theta = 1/2$. They use a combination of the explicit and implicit one-step integrators. However, they differ from a predictor-corrector method in that the component integrators are not both applied over the full interval h. The two-step midpoint algorithm has local truncation error of the order, h^3, rather than the truncation error of order h^2 of the other asymptotic algorithms, but only if $\theta = 1/2$. The one-step explicit and implicit asymptotic algorithms have the same order of local truncation error, h^2, as the corresponding linear methods.

Evaluation of the stability of nonlinear algorithms applied to nonlinear, ordinary differential equations is a difficult problem. Each of these asymptotic algorithms is A-stable. This in itself is an improvement over many linear algorithms. Conditions have been found under which the asymptotic algorithms are contractive, which is often taken as the definition of stability. The methods are not uniformly stable under the strongest definitions of stability. The range of values of the time step, h, that give stable behavior depends on the form in which the ordinary differential equation is written. In the case that $a(x) \equiv 0$, they are uniformly stable in the sense that $|x_{n+1}| \leq |x_n|$.

The algorithms allow larger step sizes within the stable range near the steady state for differential equations that exhibit asymptotic behavior. As desired, the asymptotic algorithms behave comparably to a differential equation with a globally asymptotic steady-state. The algorithms will also be shown, in a subsequent paper, to be more than competitive with linear multi-step methods when applied to stiff systems of equations.

Acknowledgement

This work was supported in part by NASA grant NAG3-1321.

References

Dahlquist, G., 1963, "A special stability problem for linear multistep methods", *BIT*, Vol. 3, pp. 27 - 43.

Dennis, S.C.R., 1959, "The Numerical Integration of Ordinary Differential Equations Possessing Exponential Type Solutions," *Proceedings of the Cambridge Philosophical Society,* Vol. 56, pp. 240–246.

Freed, A.D., Yao, M. and Walker, K.P., 1992, "Asymptotic Integration Algorithms for First-Order ODEs With Application to Viscoplasticity," (NASA TM 105587, Cleveland) *Recent Advances in Damage Mechanics in Plasticity,* J.W. Ju, ed., ASME, New York, AMD-Vol. 132 & MD-Vol. 30, pp. 49–62.

Hairer, E. and Wanner, G., 1991, *Solving Ordinary Differential Equations II,* Springer-Verlag, New York.

Henrici, P., 1962, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley and Sons, Inc., New York.

Hogge, M. A., 1978, "Accuracy and Cost of Integration Techniques for Non-linear Heat Transfer", *Second World Congress on Finite Element Methods*, Bournemouth, UK, pp. 135–154.

Hughes, T.J.R., 1983, "Analysis of Transient Algorithms with Particular Reference to Stability Behaviour," *Computational Methods for Transient Analysis,* T. Belytschko and T.J.R. Hughes, eds., North-Holland, Amsterdam, pp. 67–155.

Lambert, J. D., 1991, *Numerical Methods for Ordinary Differential Systems*, John Wiley and Sons, Ltd, Chichester.

Lichtenberg, A.J., and Lieberman, M.A., 1992, *Regular and Chaotic Dynamics*, Appl. Math. Sci., Vol 38, Springer-Verlag, New York.

Marsden, J.E., 1992, *Lectures on Mechanics*, Cambridge University Press, Cambridge.

Quinney, D., 1987, *An Introduction to the Numerical Solution of Differential Equations*, Research Studies Press, Ltd, Letchworth, Hertfordshire, England and John Wiley and Sons, Inc., New York.

Rall, L.B., 1974, "A Note on the Convergence of Newton's Method," *SIAM Journal of Numerical Analysis,* Vol. 11, pp. 34–36.

Rosinger, E. E., 1980, "Stability and convergence for non-linear difference schemes are equivalent", *J. Inst. Maths Applics*, Vol. 26, pp 143 - 149.

Walker, K.P. and Freed, A.D., 1991, "Asymptotic Integration Algorithms for Nonhomogeneous, Nonlinear, First Order, Ordinary Differential Equations," NASA TM 103793, Cleveland.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE November 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE

Nonlinear Asymptotic Integration Algorithms for One-Dimensional Autonomous Dissipative First-Order ODEs

5. FUNDING NUMBERS

WU-505-63-5A

6. AUTHOR(S)

Henry W. Haslach, Jr., Alan D. Freed, and Kevin P. Walker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E-9233

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-106780

11. SUPPLEMENTARY NOTES

Henry W. Haslach, Jr., Department of Mechanical Engineering, University of Maryland, College Park, Maryland 20742 (work funded by NASA Contract NAG3-1321); Alan D. Freed, NASA Lewis Research Center; and Kevin P. Walker, Engineering Science Software, Inc., Smithfield, Rhode Island 02917. Responsible person, Alan D. Freed, organization code 5110, (216) 433-8747.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Nonlinear asymptotic integrators are applied to one-dimensional, nonlinear, autonmous, dissipative, ordinary differential equations. These integrators, including a one-step explicit, a one-step implicit, and a one- and a two-step midpoint algorithm, are designed to follow the asymptotic behavior of a system approaching a steady state. The methods require that the differential equation be written in a particular asymptotic form. This is always possible for a one-dimensional equation with a globally asymptotic steady state. In this case, conditions are obtained to guarantee that the implicit algorithms are well defined. Further conditions are determined for the implicit methods to be contractive. These methods are all first order accurate, while under certain conditions the midpoint algorithms may also become second order accurate. The stability of each method is investigated and an estimate of the local error is provided.

14. SUBJECT TERMS Numerical Integration; Numerical stability; Differential Equations	15. NUMBER OF PAGES 28
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT